Syntax Analysis - Parsing

66.648 Compiler Design Lecture (01/28/98)

Computer Science

Rensselaer Polytechnic

Lecture Outline

Syntax Analysis and Context Free Grammars

Bottom-up Parsing

Administration

Syntax Analysis

Reading: We are currently in Chapter 4 of the text book. Please read the material and work the exercises.

Syntax Analysis:

tokens $\xrightarrow{\text{PARSER}}$ Parse Tree

Parse Tree depicts the Syntactic Structure of the input Program. Parser is a program that converts the tokens into a Parse tree.

Context Free Grammars

CFG is a notation used to specify permissible syntactic structures of a programming language.

This grammar formalizes syntactic information often presented as “railroad diagrams” in programming language specifications.

They are also referred to as Backus Normal Form (BNF) grammar.

CFG Cont...

Examples of Context Free Grammars:

E $\longrightarrow$ E + T | T| E - T

T $\longrightarrow$ T * F | F| T/F

F $\longrightarrow$ (E) | id| -E|num

E stands for expressions, T stands for terms and F stands for factors.

This grammar also takes care of precedence of operators.

CFG Cont...

Another possibilty is to write for Expressions:

E $\longrightarrow$ E+E|E-E|E*E|E/E|(E)|-E|id| num

Even though this grammar generates valid arithmetic expressions, it is ambiguous.

S $\longrightarrow$ if E then S else S | if E then S

Questions

1) What are the tokens in each of the grammar given in the prvious slides?

2) What is the starting symbol?

3) Where do you find the grammars for a programming language?

Definiton of CFG

A CFG G= (N,T,S,P) consists of N is a set of Nonterminal Symbols - syntactic variables

T is a set of Terminal Symbols - scanner tokens

S is a start nonterminal.

(qn: What is the starting nonterminal of the two languages we described in the earlier slides)

P is a set of productions. The productions are of the form A $\rightarrow$ alpha, where alpha is a string of terminals and nonterminals.

More on CFG

(Please recollect what is the difference between regular grammar and context free grammar - in terms of productions)

Let us look at the Context Free grammar for Java.

What is the starting nonterminal?

What are the productions for statement?

(Pages in the Language Specification Book)

CFG Cont...

A string of terminals w is a sentence of G, if there exists a derivation sequence of n >=1 steps of the form

S (start) = x_0 ==>x_1==>x_2… ==>x_n=w.

For example compiler+is*fun is a valid sentence in the expression grammar.

Each derivation step represents a single rewrite and must have the form x_j = u V p ==> u b p, where there is a production of the form V = b. We call u b p = x_{j+1}

The language denoted by G is the set,

L(G) = { w | w is a sentence.}

Syntax Analysis Problem

Find a derivation sequence in grammar G for a given input stream of tokens. (or say if none exist). If a derivation exists, then we say that given input tokens is syntactically correct or it is a syntax error.

Rightmost derivation sequence: a derivation sequence in which the rightmost nonterminal is replaced at each step.

Syntax Analysis Problem cont...

One can define leftmost derivation analogously.

Of course, when we are replacing the rightmost nonterminal, say L, we do not know which of the productions in which L appears on the left hand side to apply.

In each step of a rightmost derivation, the string of symbols right of the rightmost nonterminal is a string over terminal symbols.

Expression Grammar - Examples

Rightmost derivation for 19+97*8.9

E ==> E + T

==> E + T * F

==> E + T * num

==> E + F * num

==> E + num * num

==> T + num * num

==> F + num * num ==> num + num * num

Parse Trees

A parse tree is a graphical represntation of a sentential form (what is the difference between a sentence and a sentential form).

Nodes of a tree represent grammar symbols (nonterminals or terminals) and tree edges represent a derivation step.

Parse Tree

Draw a parse tree for 19 + 98 * 8.9

Draw a parse tree - x + 7

Ambiguous Grammars

A grammar G is ambiguous iff G can produce more than one rightmost derivation sequence (i.e. more than one parse tree) for some sentence in L(G).

For efficient parsing and semantic analysis, it is desirable to replace an ambigous grammar by an equivalent unambiguous grammar G' such that L(G) = L(G')

Administration

We are in Chapter 4 of Aho, Sethi and Ullman's book. Please read that chapter and chapters 1, 2 and 3.

Work out the unstarred exercises of chapter 3 and first few problems in 4.

Lex and Yacc Manuals are handed out. Please read them.

First Project is in the web.

It consists of three parts.

1) To write a lex program

2) To write a YACC program.

3) To write five sample Java programs. They can be either applets or application programs

Comments and Feedback

Please let me know if you have not found a project partner.